DETAILED ACTION

1. Applicant's amendment dated March 20, 2009, responding to the Office Action mailed October 20, 2008 provided in the rejection of claims 1-23.

Claims 1-23 remain pending in the application and which have been fully considered by the examiner.

Examiner called Ann M. McCrackin (Reg. No. 42,858) on May 27, 2009 and Ms. McCrackin confirmed that the RCE filing dated March 20, 2009 was an inadvertent mistake. Thus, the Office Action will treat it as a regular amendment.

Further, Applicant's arguments with respect to claims have been fully considered but are not persuasive, thus the previous rejections are maintained and reproduced below. Please see the section of "Response to Arguments" for details.

2. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory

action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Information Disclosure Statement

3. The information disclosure statement (IDS) submitted on March 20, 2009 was filed after the mailing date of the Office action on October 20, 2008. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Claim Rejections – 35 USC § 102(b)

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102(b) that form the basis for the rejections under this section made in this office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1-23 are rejected under 35 U.S.C. 102(b) as being anticipated by Quinn Able Jacobson (*High-Performance Frontends for Trace Processors, 1999, University of Wisconsin - Madison*) (hereinafter 'Jacobson')

5. **As to claim 1** (Previously Presented), Jacobson discloses an apparatus comprising:

- a processing unit of a processor;

- a memory coupled to the processor; and
- an instruction set operable on the processing unit of the processor and including instruction;
- to instantiate a data structure to collect a representation of a working set (e.g., P. 167, Sec. 5.2.2 Implementing a predictor, 2nd Para - ... along with PC of the current branch instruction (i.e., a branch working set) to index a table ... to combine the PC and the global branch outcomes to produce ...; P. 168, Fig. 5-3 Example global history branch predictor, elements of 'Table of 2-bit counters'; 'Current Branch PC'; P. 167, last Para - ... built around a table of 2-bit counters ...; P. 165, Sec. 5.2.1.1, Lines 8-14; Sec. 5.3.1 - Corrected Predictor; Sec. 5.3.2 – Hybrid Predictor); and
- defining a hash unit operable on the processing unit to map a plurality of working set elements into the data structure using a hash function (e.g., P. 168, Fig. 5-3 Example global history branch predictor, element of 'Hash Funct'; Fig. 5-5 – Hashing Function; P. 170, last Para - ... The hashing function combines the trace starting address and branch outcomes into a condensed encoding (see Fig. 5-5))

6. **As to claim 2** (Previously Presented) (incorporating the rejection in claim 1), Jacobson discloses the apparatus wherein the data structure is a 2^n x m bit table, where n is a number of bit table entries and m is a width of the bit table (e.g., P. 168, Fig. 5-3 Example global history branch predictor, element of 'Table of 2-bit counters')

7. **As to claim 3** (Original) (incorporating the rejection in claim 2), Jacobson discloses the apparatus wherein m is in the range of 1 to 64 (e.g., P. 64, Lines 1-4 - ... an aggregate instruction window size of 64 instructions ...)

8. **As to claim 4** (Original) (incorporating the rejection in claim 2), Jacobson discloses the apparatus wherein m = 1 (e.g., P. 168, Fig. 5-3 Example global history branch predictor, element of 'Table of 2-bit counters')

9. **As to claim 5** (Original) (incorporating the rejection in claim 2), Jacobson discloses the apparatus wherein n is in the range of 1 to 20 (e.g., P. 168, Fig. 5-3 Example global history branch predictor, element of 'Table of 2-bit counters')

10. **As to claim 6** (Original) (incorporating the rejection in claim 1), Jacobson discloses the apparatus wherein the data structure is a 2^n-bit vector (e.g., P. 168, Fig. 5-3 Example global history branch predictor, element of 'Table of 2-bit counters')

11. **As to claim 7** (Original) (incorporating the rejection in claim 6), Jacobson discloses the apparatus wherein n = 1 (e.g., P. 168, Fig. 5-3 Example global history branch predictor, element of 'Table of 2-bit counters')

12. **As to claim 8** (Original), Jacobson discloses a computerized method of creating a representation of a working set, the computerized method comprising:

- mapping a plurality of working set elements into fields of a data structure using a hash function (e.g., P. 167, Sec. 5.2.2 Implementing a predictor, 2nd Para - ... along with PC of the current branch instruction (i.e., a branch working set) to index a table ... to combine the PC and the global branch outcomes to produce ; P. 168, Fig. 5-3 Example global history branch predictor, elements of 'Table of 2-bit counters'; 'Current Branch PC'; P. 167, last Para - ... built around a table of 2-bit counters ...; P. 165, Sec. 5.2.1.1, Lines 8-14; Sec. 5.3.1 - Corrected Predictor; Sec. 5.3.2 – Hybrid Predictor; P. 168, Fig. 5-3 Example global history branch predictor, element of 'Hash Funct'; Fig. 5-5 – Hashing Function; P. 170, last Para - ... The hashing function combines the trace starting address and branch outcomes into a condensed encoding (see Fig. 5-5))

13. **As to claim 9** (Original) (incorporating the rejection in claim 8), Jacobson discloses the computerized method wherein the mapping is performed for a fixed interval of program execution (e.g., P. 93, 3rd Para - ... periodically interrupted and all the dynamic structures (caches and predictors) are flushed ...)

14. **As to claim 10** (Original) (incorporating the rejection in claim 9), Jacobson discloses the computerized method wherein the data structure is reset prior to

each fixed interval of program execution (e.g., P. 60, Lines 1-2 – … the entire buffer can be cleared by simply resetting the pointer to the head of the buffer)

15. **As to claim 11** (Original) (incorporating the rejection in claim 10), Jacobson discloses the computerized method further comprising saving the fields of the data structure prior to resetting the data structure (e.g., P. 10, Lines 3-4 – by saving path history information …)

16. **As to claim 12** (Original), Jacobson discloses a computerized method of creating a representation of a working set, the computerized method comprising:

- executing a program for a fixed interval, the program comprising instructions identified by a program counter (e.g., P. 168, Fig. 5-3 - Example global history branch predictor, elements of 'Table of 2-bit counters'; 'Current Branch PC');
- performing a hash function on the program counter to create a hash value for each instruction executed during the fixed interval (e.g., P. 168, Fig. 5-3 Example global history branch predictor, element of 'Hash Funct'; Fig. 5-5 – Hashing Function; P. 170, last Para - ... The hashing function combines the trace starting address and branch outcomes into a condensed encoding (see Fig. 5-5)); and
- updating a field of a table indexed by the hash value wherein the table represents the working set (e.g., P. 167, Sec. 5.2.2 Implementing a predictor, 2nd Para - … along with PC of the current branch instruction

(i.e., a branch working set) to index a table … to combine the PC and the global branch outcomes to produce …; P. 168, Fig. 5-3 Example global history branch predictor, elements of 'Table of 2-bit counters'; 'Current Branch PC'; P. 167, last Para - … built around a table of 2-bit counters …; P. 171, 2nd full Para through 1st Para - … The index generation mechanism uses a few bits from ….)

17. **As to claim 13** (Original), Jacobson discloses a computer system comprising:

- a bus;
- a memory coupled to the bus; and
- a processor coupled to tile memory and the bus; the processor comprising:
  - a data structure to collect a representation of a working set (e.g., P. 167, Sec. 5.2.2 Implementing a predictor, 2nd Para - … along with PC of the current branch instruction (i.e., a branch working set) to index a table … to combine the PC and the global branch outcomes to produce …; P. 168, Fig. 5-3 - Example global history branch predictor, elements of 'Table of 2-bit counters'; 'Current Branch PC'; P. 167, last Para - … built around a table of 2-bit counters …; P. 165, Sec. 5.2.1.1, Lines 8-14; Sec. 5.3.1 - Corrected Predictor; Sec. 5.3.2 – Hybrid Predictor); and

- a hash unit to map a plurality of working set elements into the data structure using a hash function (e.g., P. 168, Fig. 5-3 Example global history branch predictor, element of 'Hash Funct'; Fig. 5-5 – Hashing Function; P. 170, last Para - ... The hashing function combines the trace starting address and branch outcomes into a condensed encoding (see Fig. 5-5))

18. **As to claim 14** (Previously Presented) (incorporating the rejection in claim 13), Jacobson discloses the computer system further comprising:

- an instruction retirement unit; and
- wherein the data structure and the hash unit are part of an instruction retirement unit (e.g., Fig. 1-1 – Typical processor organization, element of 'Instruction Retirement Pipeline'; P. 3, 1st full Para)

19. **As to claim 15** (Original), Jacobson discloses a computerized method of estimating size of a working set, the method comprising:

- Receiving a signature for a working set (e.g., P. 167, Sec. 5.2.2 Implementing a predictor, 2nd Para - … along with PC of the current branch instruction (i.e., a branch working set) to index a table … to combine the PC and the global branch outcomes to produce …; P. 168, Fig. 5-3 - Example global history branch predictor, elements of 'Table of 2-bit counters'; 'Current Branch PC'; P. 167, last Para - … built around a table of 2-bit counters …); and

- Estimating the size of the working set based on the size of the signature (e.g., P. 53, 1st Para - … with trace pre-construction based on dynamic learning … working set size ...)

20. **As to claim 16** (Original) (incorporating the rejection in claim 15), Jacobson does not disclose the computerized method wherein the estimating is performed with the following function:

$K \qquad K = \log(1-f) \Big/ \log\left(1 - \frac{1}{2^n}\right),$

wherein K is the number of unique working set elements, 2^n is the number of entries in the signature, and f is the fraction of 1 's in the signature.

However, it is well known in the art of mathematical prediction equations to incorporate a logarithm of the probability of transitioning multipliers into equations in order to obtain the benefits know in the art.

21. **As to claim 17** (Original), Jacobson discloses a computerized method of detecting working set changes, the method comprising:

- comparing a current working set signature to a previous working set signature (e.g., P. 167, Sec. 5.2.2 Implementing a predictor, 2nd Para - … along with $K = \log(1-f) \Big/ \log\left(1 - \frac{1}{2^n}\right),$ ion (i.e., a branch working set) to index a ne global branch outcomes to produce …; P. 16, 2nd Para - … predict the outcome of branches based on

previous branch behavior ... a Pattern History Table ... containing two-bit saturating counter ...);

- calculating a relative signature distance between the current working set signature and the previous working set signature (e.g., P. 61, 2nd Para - ... along with each instruction is kept its minimal distance from the first instruction of the region ... by looking at the minimal distance of a trace start point and the lowest minimal distance on the worklist ... to make a conservative decision ...); and
- identify a working set change when the relative signature distance exceeds a predetermined threshold (e.g., P. 16, 2nd Para - ... If the counter's value is above some threshold the branch is predicted taken ...)

22. **As to claim 18** (Original) (incorporating the rejection in claim 17), Jacobson discloses the computerized method wherein the working set change indicates a phase change in a program (e.g., P. 168, Fig. 5-3 Example global history branch predictor, elements of 'Table of 2-bit counters'; 'Current Branch PC'; P. 167, last Para - ... built around a table of 2-bit counters ...; P. 165, Sec. 5.2.1.1, Lines 8-14; Sec. 5.3.1 - Corrected Predictor; Sec. 5.3.2 – Hybrid Predictor)

23. **As to claim 19** (Original), Jacobson discloses a computerized method of identifying a recurring working set, the method comprising:

- comparing a current working set signature to one or more previous working set signatures (e.g., P. 16, 2nd Para - ... predict the outcome of branches based on previous branch behavior ... a Pattern History Table ... containing two-bit saturating counter ...);
- calculating a relative signature distance between the current working set signature and the one or more previous working set signatures (e.g., P. 61, 2nd Para - ... along with each instruction is kept its minimal distance from the first instruction of the region ... by looking at the minimal distance of a trace start point and the lowest minimal distance on the worklist ... to make a conservative decision ...); and
- identifying a recurring working set when the relative signature distance between the current working set signature and one of the previous working set signatures is within a predetermined threshold (e.g., P. 16, 2nd Para - ... If the counter's value is above some threshold the branch is predicted taken ...)

24. **As to claim 20** (Original) (incorporating the rejection in claim 19), Jacobson discloses the computerized method further comprising identifying a new working set when the relative signature distance between the current working set signature the one or more previous working set signatures exceeds a predetermined threshold (e.g., P. 16, 2nd Para - ... If the counter's value is above some threshold the branch is predicted taken ...)

25. **As to claim 21** (Original) (incorporating the rejection in claim 20), Jacobson discloses the computerized method further comprising maintaining a table of the one or more previous working set signatures (e.g., P. 16, 2nd Para - … a Pattern History Table (PHT) …)

26. **As to claim 22** (Original), Jacobson discloses a hardware reconfiguration method comprising:

- maintaining a table comprising a plurality of working set signatures for a program (e.g., ., P. 168, Fig. 5-3 Example global history branch predictor, elements of 'Table of 2-bit counters'; 'Current Branch PC'; P. 167, last Para - … built around a table of 2-bit counters …; P. 165, Sec. 5.2.1.1, Lines 8-14; Sec. 5.3.1 - Corrected Predictor; Sec. 5.3.2 – Hybrid Predictor);
- upon detecting a working set change, looking up a working set signature for a current working set in the table (e.g., P. 174, 1st full Para - The larger counter is used to enable the detection of very consistent behavior …);
- if the working set signature is in the table, reinstating a hardware configuration for the current working set; and
- if the working set signature is not in the table; identifying a new hardware configuration for the current working set and saving the working set signature and the new hardware configuration (e.g., P. 16, 2nd Para - … predict the outcome of branches based on previous branch behavior … a Pattern History Table … containing two-bit saturating counter …; Abstract,

4th Para - ... takes advantage of the trace cache to dynamically optimize applications ... take advantage of implementation-specific hardware ...;)

27. **As to claim 23** (Original) (incorporating the rejection in claim 22), Jacobson discloses the method wherein the working set change indicates a phase change (e.g., P. 16, 2nd Para - ... predict the outcome of branches based on previous branch behavior ... a Pattern History Table ... containing two-bit saturating counter ...)

Response to Arguments

28. Applicant's arguments filed on March 20, 2009 have been fully considered, but they are not persuasive.

In the remarks, Applicant argues that, for examples:

(A.1) Jacobson's traces do not teach or disclose "a working set" or "a plurality of working set elements" (recited in REMARKS, on page 7, second half portion; page 8, last full paragraph); "a data structure" to "collect a representation of a working set"; and "map a plurality of working set elements into the data structure" (recited in REMARKS, on page 8, first full paragraph; page 9, second full paragraph) (emphasis added)

Examiner's response:

(R.1) Examiner disagrees.

Firstly, the limitations, argued by Applicant, of "a working set" or "a plurality of working set elements"; "a data structure" to "collect a representation of a working set"; and "map a plurality of working set elements into the data structure" are clearly illustrated in Exhibit A below with annotated dashed circle and lines (emphasis added)

Secondly, in light of the specification, the specification recites "A working set can be associated with a specific type of memory access, or some combination ... As another example, a working set can correspond to fetched branch instructions, in which case it is referred to as a branch working set" (lines 14-19, page 3 – emphasis added). Further, Jacobson discloses instantiating a data structure to collect a representation of a working set (e.g., P. 167, Sec. 5.2.2 Implementing a predictor, 2nd Para - ... along with PC of the current branch instruction (i.e., a branch working set) to index a table ... to combine the PC and the global branch outcomes to produce ...; P. 168, Fig. 5-3 Example global history branch predictor, elements of 'Table of 2-bit counters'; 'Current Branch PC'; P. 167, last Para - ... built around a table of 2-bit counters ...; P. 165, Sec. 5.2.1.1, Lines 8-14; Sec. 5.3.1 - Corrected Predictor; Sec. 5.3.2 – Hybrid Predictor); and defining a hash unit operable on the processing unit to map a plurality of working set elements into the data structure using a hash function (e.g., P. 167, Sec. 5.2.2 Implementing a predictor, 2nd Para - ... along with PC of the current branch instruction (i.e., a branch working set) to index a table ... to combine the PC and the global branch outcomes to produce ...; 168, Fig. 5-3 Example global history branch predictor, element of 'Hash Funct'; Fig. 5-5 –

Hashing Function; P. 170, last Para - ... The hashing function combines the trace starting address and branch outcomes (i.e., a branch working set) into a condensed encoding (see Fig. 5-5))

Thirdly, however, the characteristic of the claimed invention, as argued in the REMARKS (page 7), excerpts as follows:

> "In a program, a working set $W(t_i, \tau)$ for i=1,2..., is a set of distinct memory segments $\{s_1, s_2 \ldots s_\omega\}$ accessed over the i^{th} window of size τ. . ."[7]
>
> With regard to this definition, the applicant's specification further states that "[t]he window is a sequence of τ consecutive memory accesses. The working set size is ω, the cardinality of the set of unique segments that are accessed by members of the window. In one example, the segments are memory regions of some fixed size, such as a page. In another example, the segments are memory regions of the size of a cache memory block."[8] A cache memory block is also a fixed size region.

are not contained in the claim limitations (emphasis added)

Furthermore, in response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., 'a working set' definition or definition of 'a plurality of working set elements') are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

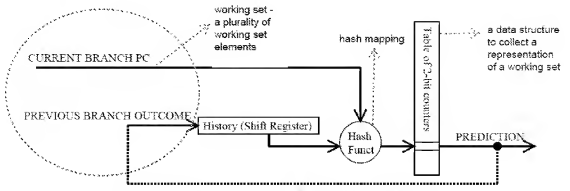

Figure 5-3 Example global history branch predictor.

(Exhibit A - Fig. 5-3, on page 168, Jacobson's reference)

Conclusion

29. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/
Ben C. Wang
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192